



Tipo double

Transcripción

[00:00] Ahora, ¿qué pasaría si yo no necesito trabajar con enteros sino preciso trabajar con valores decimales? ¿Podré seguir usando el tipo de dato int? Por ejemplo, ¿podría yo asignar edad igual 30.45? Nos está dando un error de compilación aquí. ¿Por qué es? Porque como lo sugiere el mismo nombre de la variable, solo soporta valores enteros.

[00:29] Si nosotros queremos un valor decimal, necesitamos usar otro tipo de variable, y en este caso el tipo de variable es `double`. Vamos a borrar esto y vamos a comenzar trabajando con `double` en este caso. ¿Cómo es `double`? `Double` salario igual 1250.50. En este caso veo que compila correctamente.

[01:06] Y si queremos imprimir este valor, ponemos aquí directamente `salario`, salvamos, ejecutamos y vemos que él imprime 1250.5. ¿Por qué él está ignorando el 0 al final? Porque no tiene ningún valor aritmético en este caso. Si fuera por ejemplo un 56, ejecutamos y él ya lo va a tomar en cuenta. Perfecto.

[01:53] Y ahora, ¿será que podemos asignar un valor entero en ese valor decimal, por ejemplo un `double` edad, como el caso anterior, igual a 28? Podemos. ¿Por qué? Porque el espacio de bits que tiene un tipo de dato `double` es justamente eso, es el doble de espacio de bits que tiene un tipo de dato `int`.

[02:19] Digamos que tenemos mucho más espacio en esta variable para poder asignar valores numéricos, por lo cual 28 como número entero entra tranquilamente aquí en un tipo de dato `double`, cosa que ya vimos, que si por ejemplo tenemos un `int` salario, no puede tener asignado un valor decimal.

Perfecto. En este caso me marcó un error en salario porque yo ya declaré un nombre salario aquí.

[03:01] Recordemos que Java no me permite declarar dos veces la misma variable. Yo puedo sobrescribir el valor de la variable cuantas veces quiera pero no puedo declarar ella dos veces dentro del mismo contexto, en este caso el contexto del método. Borramos esto de aquí.

[03:21] Entre las demás operaciones que nos permiten ejecutar double, podemos hacer incluso divisiones por ejemplo. Double salarioMitad va a ser igual a salario entre dos. ¿Se dieron cuenta aquí que yo ya no tuve que rescribir nuevamente el valor numérico de salario sino que yo directamente invoqué a la variable y lo dividí entre dos?

[03:54] Vamos a ver qué resultado nos da. En este caso vamos a imprimir salarioMitad. Salvamos aquí, damos run y tenemos aquí salarioMitad 625.28. ¿Entonces será que si usamos un tipo de dato entero él no soporta división? Él soporta división, de hecho que soporta.

[04:20] Por ejemplo vamos a int división. Y vamos a dividir este mismo valor, en este caso ya sabemos que no podemos asignar un valor doble a int entonces vamos a darle el valor entero entre 3 por ejemplo. Yo no sé cuánto va a dar esto pero yo asumo que va a dar algún decimal.

[04:46] Y si imprimimos esto, ¿será que él va a asignar un valor decimal a int? ¿Será que el lenguaje cometerá alguna inconsistencia? Vamos a averiguarlo. Vamos aquí, guardamos y vemos que él dio 416. ¿Qué es lo que está pasando aquí?

[05:08] Que él, como es un tipo de dato entero, él está redondeando el valor de la división a un valor entero, por lo cual nosotros estaríamos perdiendo exactitud decimal al asignar una división dentro de un campo entero. Es así como funciona los tipos de datos int y double.

[05:31] Ahora, otro dato curioso, por ejemplo. ¿Será que yo puedo imprimir división antes de declarar la variable? No puedo. ¿Por qué? Porque otro concepto de variables en Java es que yo no puedo usar ellas hasta no haberlas declarado primero. Por ejemplo, yo podría hacer algo como esto.

[05:56] Hacer solo declarar división, punto y coma, declararé la variable, y aquí abajo asignar el valor de división. En este caso él me está dando un error aquí porque división en este punto no tiene nada, no ha sido inicializada, yo estoy inicializando ella aquí abajo.

[06:15] Entonces yo no puedo usar ninguna variable si ya antes no he inicializado ella. Esta expresión de aquí es un poco ambigua porque tranquilamente, como he venido haciendo a lo largo de todos los ejercicios puedo asignar directamente el valor de división aquí en una sola línea.

[06:38] Entonces con esto ya tenemos dos tipos de variable, ya sabemos cómo usarlo, ya sabemos para qué casos nos pueden ser útiles y en qué casos podemos encontrar ciertas inconsistencias para obtener resultados, como por ejemplo asignar el resultado decimal de una división a un tipo entero.